

KARTA PRZEDMIOTU

Kod przedmiotu	0613-2INF-C22-PRO	
Nazwa przedmiotu w języku	polskim	<i>Programowanie obiektowe</i> <i>Object oriented programming</i>
	angielskim	

1. USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

1.1. Kierunek studiów	Informatyka
1.2. Forma studiów	stacjonarne
1.3. Poziom studiów	studia I-stopnia inżynierskie
1.4. Profil studiów	ogólnoakademicki
1.5. Osoba przygotowująca kartę przedmiotu	Mirosław Głowacki
1.6. Kontakt	głowacki@metal.agh.edu.pl

2. OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

2.1. Język wykładowy	polski
2.2. Wymagania wstępne	Języki i środowisko programisty Wstęp do programowania Metody programowania

3. SZCZEGÓŁOWA CHARAKTERYSTYKA PRZEDMIOTU

3.1. Forma zajęć	wykłady, ćwiczenia laboratoryjne	
3.2. Miejsce realizacji zajęć	zajęcia w pomieszczeniu dydaktycznym UJK	
3.3. Forma zaliczenia zajęć	wykłady – egzamin ćwiczenia laboratoryjne – zaliczenie z oceną	
3.4. Metody dydaktyczne	wykład, zajęcia laboratoryjne przy komputerach	
3.5. Wykaz literatury	podstawowa	<ol style="list-style-type: none"> 1. B. Stroustrup, Język C++. Kompendium wiedzy, Wyd. IV, Helion 2014 2. S. Prata. Język C++. Szkoła programowania. Wydanie VI, Helion 2013 3. Nowak R., Pająk A., Język C++. Mechanizmy, wzorce, biblioteki, BTC, Legionowo 2010 4. Lisowski E., Filo G., Metodyka programowania obiektowego z przykładami w C++, Wyd. Politechniki Krakowskiej, Kraków 2009 5. J. Grębosz, Symfonia C++, Oficyna Kallimach, Kraków 2006 6. J. Grębosz, Pasja C++, Oficyna Kallimach, Kraków
	uzupełniająca	<ol style="list-style-type: none"> 1. B. Meyer, Programowanie zorientowane obiektowo, Helion, Gliwice 2005 2. McLaughlin B. D., Pollice G., West D., Rusz głową. Analiza i projektowanie obiektowe, Helion, Gliwice 2010 3. Shalloway A., Trott J. T., Projektowanie zorientowane obiektowo. Wzorce projektowe, Helion, Gliwice 2005 4. R. Sedgewick, Algorytmy w C++, Oficyna Wyd. RM, 1999 5. C. Dellanoy, Ćwiczenia z języka C++: Programowanie obiektowe, WNT, Warszawa, 1993

4. CELE, TREŚCI I EFEKTY UCZENIA SIĘ

4.1. Cele przedmiotu

Wykład:

- C1. Celem wykładu jest zapoznanie studentów z paradygmatem programowania obiektowo zorientowanego. Studenci poznają programowanie zgodne z postrzeganiem rzeczywistości, kryteria obiektowości oraz techniki programowania obiektowego. Wykład poświęcony jest zarówno obiektowemu modelowaniu dziedziny jak i pojęciom oraz ogólnym zasadom projektowania i programowania obiektowego.
- C2. Równolegle przedstawiane są mechanizmy programistyczne umożliwiające implementację modelu obiektowego w oparciu o język C++. W trakcie kolejnych wykładów omawiane są: klasy, tryb dostępu do składników klas, konwersje, przeciążenia operatorów, dziedziczenie polimorfizm i generyczność.

Ćwiczenia laboratoryjne:

- C1. W trakcie ćwiczeń laboratoryjnych studenci poznają aspekty programowania obiektowo zorientowanego od strony praktycznej wykonując niewielkie zadania zlecane przez prowadzących zajęcia. W ramach prac własnych wykonują oni samodzielny projekt i jego implementację bazując na informacjach uzyskanych w trakcie wykładu i ćwiczeń laboratoryjnych.

4.2. Treści programowe

Wykład

Przegląd podstawowych paradygmatów programowania. Środowiska programistyczne zorientowane obiektowo. Pojęcie obiektu, proste przykłady obiektów, analogia do obiektów rzeczywistych. Obiektowe modelowanie dziedziny. Cechy programowania obiektowego. Typy obiektów, typy definiowane przez użytkownika, statyczna kontrola typów. Cechy obiektów - pola i metody. Klasy i kapsułkowanie - klasy jako typy, składniki klas, wskaźnik *this*, konstruktory i destruktory, statyczne składniki klas, pola stałe, pola wskaźnikowe i referencyjne. Tryby dostępu do składników klas, ukrywanie informacji, funkcje zaprzyjaźnione. Przestrzenie nazw. Automatyczna konwersja typów. Operatory i ich przeciążenia. Dziedziczenie klas, dostęp do składników klas podstawowych, dziedziczenie a zawieranie klas, ponowna definicja cech. Dziedziczenie wielokrotne i wielopokoleniowe - ryzyko wieloznaczności, klasy wirtualne. Wskaźniki do instancji klas, niejawna konwersja typów, wiązanie dynamiczne, funkcje wirtualne i polimorfizm, klasy abstrakcyjne. Wyjątki i ich obsługa. Generyczność - szablony funkcji i klas, parametry szablonów, klasy specjalizowane, ograniczona generyczność.

Ćwiczenia laboratoryjne

Zintegrowane środowisko programistyczne zorientowane obiektowo. Porównanie paradygmatów programowania proceduralnego i obiektowo zorientowanego w języku C++ - obiekty jako parametry funkcji. Definiowanie klas i ich pól o dostępie publicznym. Metody klas, deklarowanie i definiowanie metod wewnątrz i na zewnątrz klasy. Tworzenie instancji klas – przykłady prostych obiektów, rola wskaźnika *this*, odwzorowanie obiektów rzeczywistych – obiekty geometryczne. Tworzenie klas ze składnikami prywatnymi, ukrywanie i wybiórcze udostępnianie informacji, dostęp do prywatnych pól klasy – metody publiczne i funkcje zaprzyjaźnione. Praca z konstruktorami i destruktorami, definiowanie własnych konstruktorów, inicjalizacja pól stałych i referencyjnych, definiowanie liczników i tablic dynamicznych – rola konstruktorów i destruktorów. Korzystanie z automatycznej konwersji typów, definiowanie operatorów konwersji i konstruktorów konwertujących. Przeciążanie operatorów. Obiektowe modelowanie dziedziny - zaprojektowaniu i implementacja prostego programu obiektowo zorientowanego. Korzystanie z mechanizmu dziedziczenia klas, uzyskiwanie dostępu do składników klas bazowych i klas zawartych, tworzenie list inicjalizacyjnych konstruktorów klas pochodnych. Usuwanie problemów związanych z dziedziczeniem wielokrotnym, tworzenie klas wirtualnych. Używanie wskaźników do instancji klas, poznanie mechanizmu niejawnej konwersji typów i wiązania dynamicznego. Tworzenie funkcji wirtualnych, korzystanie z polimorfizm, tworzenie klas abstrakcyjnych. Tworzenie aplikacji z obsługą wyjątków. Budowa programów generycznych – tworzenie szablonów klasy i metod, korzystanie z parametrów typu i parametrów pozatypowych.

4.3. Przedmiotowe efekty uczenia się

Efekt	Student, który zaliczył przedmiot	Odniesienie do kierunkowych efektów uczenia się
w zakresie WIEDZY:		
W01	zna podstawy teorii programowania obiektowego, rozumie sposób podejścia do obiektów oraz ich metod i pól, a także sposoby wzajemnego powiązania klas obiektów i komunikowania się poszczególnych obiektów	INF1A_W08
W02	definiuje hierarchię klas, przekazywanie komunikatów pomiędzy obiektami oraz zaprojektuje sprawnie działającą aplikację opartą o teorię programowania obiektowo zorientowanego	INF1A_W08
w zakresie UMIEJĘTNOŚCI:		
U01	analizuje przedstawiony mu problem pod kątem wymagań programistycznych oraz zaprojektuje odpowiedni system rozwiązania zadań z niego wynikających	INF1A_U12 INF1A_U13 INF1A_U14
U02	ma zdolność do samodzielnego zaimplementowania aplikacji w języku C++, usunięcia błędów i przetestowania obiektowo zorientowanego kodu aplikacji	INF1A_U12 INF1A_U13 INF1A_U14
w zakresie KOMPETENCJI SPOŁECZNYCH:		
K01	jest zdolny do rozwiązywania konkretnych problemów z życia codziennego przy pomocy uzyskanej wiedzy i umiejętności praktycznych oraz umie zaproponować ich rozwiązanie z wykorzystaniem narzędzi informatycznych	INF1A_K01 INF1A_K02 INF1A_K03 INF1A_K04

4.4. Sposoby weryfikacji osiągnięcia przedmiotowych efektów uczenia się

Efekty przedmiotowe (symbol)	Sposób weryfikacji (+/-)											
	Egzamin pisemny			Kolokwium			Zadania domowe			Sprawozdania		
	Forma zajęć			Forma zajęć			Forma zajęć			Forma zajęć		
	W	L	P	W	L	P	W	L	P	W	L	P
W01	+			+								
W02	+			+								
U01								+				+
U02								+				+
K01	+							+				+

4.5. Kryteria oceny stopnia osiągnięcia efektów uczenia się		
Forma zajęć	Ocena	Kryterium oceny
wykład (W)	3	osiągnięcie <50-60) % wymogów stosowanych w metodach oceny
	3,5	osiągnięcie <61-70) % wymogów stosowanych w metodach oceny
	4	osiągnięcie <71-80) % wymogów stosowanych w metodach oceny
	4,5	osiągnięcie <81-90) % wymogów stosowanych w metodach oceny
	5	osiągnięcie <91-100> % wymogów stosowanych w metodach oceny
laboratorium (L)	3	osiągnięcie <50-60) % wymogów stosowanych w metodach oceny
	3,5	osiągnięcie <61-70) % wymogów stosowanych w metodach oceny
	4	osiągnięcie <71-80) % wymogów stosowanych w metodach oceny
	4,5	osiągnięcie <81-90) % wymogów stosowanych w metodach oceny
	5	osiągnięcie <91-100> % wymogów stosowanych w metodach oceny
projekt (P)	3	osiągnięcie <50-60) % wymogów stosowanych w metodach oceny
	3,5	osiągnięcie <61-70) % wymogów stosowanych w metodach oceny
	4	osiągnięcie <71-80) % wymogów stosowanych w metodach oceny
	4,5	osiągnięcie <81-90) % wymogów stosowanych w metodach oceny
	5	osiągnięcie <91-100> % wymogów stosowanych w metodach oceny

5. BILANS PUNKTÓW ECTS – NAKŁAD PRACY STUDENTA

Kategoria	Obciążenie studenta	
	Studia stacjonarne	Studia niestacjonarne
<i>LICZBA GODZIN REALIZOWANYCH PRZY BEZPOŚREDNIM UDZIALE NAUCZYCIELA /GODZINY KONTAKTOWE/</i>		
<i>Udział w wykładach*</i>	30	
<i>Udział w ćwiczeniach, konwersatoriach, laboratoriach*</i>	60	
<i>Udział w konsultacjach</i>	10	
<i>Udział w egzaminie/kolokwium zaliczeniowym*</i>		
<i>SAMODZIELNA PRACA STUDENTA /GODZINY NIEKONTAKTOWE/</i>		
<i>Przygotowanie do wykładu*</i>	10	
<i>Przygotowanie do ćwiczeń, konwersatorium, laboratorium*</i>	45	
<i>Przygotowanie do egzaminu/kolokwium*</i>	20	
<i>Zebrań materiałów do projektu, kwerenda internetowa*</i>		
<i>Opracowanie prezentacji multimedialnej*</i>		
<i>Inne (należy wskazać jakie? np. e-learning pod kontrolą nauczyciela)*</i>		
ŁĄCZNA LICZBA GODZIN	175	
PUNKTY ECTS za przedmiot	7	

*niepotrzebne usunąć

Przyjmuję do realizacji (data i podpisy osób prowadzących przedmiot w danym roku akademickim)

.....